

Decentralized Learning for Multi-player Multi-armed Bandits

Dileep Kalathil, Naumaan Nayyar and Rahul Jain

Abstract

We consider the problem of distributed online learning with multiple players in multi-armed bandits (MAB) models. Each player can pick among multiple arms. When a player picks an arm, it gets a reward. We consider both i.i.d. reward model and Markovian reward model. In the i.i.d. model each arm is modelled as an i.i.d. process with an unknown distribution with an unknown mean. In the Markovian model, each arm is modelled as a finite, irreducible, aperiodic and reversible Markov chain with an unknown probability transition matrix and stationary distribution. The arms give different rewards to different players. If two players pick the same arm, there is a “collision”, and neither of them get any reward. There is no dedicated control channel for coordination or communication among the players. Any other communication between the users is costly and will add to the regret. We propose an online index-based distributed learning policy called dUCB₄ algorithm that trades off *exploration v. exploitation* in the right way, and achieves expected regret that grows at most as $near-O(\log^2 T)$. The motivation comes from opportunistic spectrum access by multiple secondary users in cognitive radio networks wherein they must pick among various wireless channels that look different to different users. This is the first distributed learning algorithm for multi-player MABs to the best of our knowledge.

Index Terms

Distributed adaptive control, multi-armed bandit, online learning, multi-agent systems.

Dileep Kalathil, Naumaan Nayyar and Rahul Jain ((manisser,nnayyar,rahul.jain)@usc.edu) are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA. This research is supported by AFOSR grant FA9550-10-1-0307 and NSF CAREER award CNS-0954116.

A preliminary version of this paper is under submission to IEEE CDC 2012. This version contains proofs of all theorems as well as new results on Markovian MABs.

I. INTRODUCTION

In [1], Lai and Robbins introduced the classical non-Bayesian multi-armed bandit model. Such models capture the essence of the learning problem that players face in an unknown environment, where the players must not only *explore* to learn but also *exploit* in choosing the best arm. Specifically, suppose a player can choose between N arms. Upon choosing an arm i , it gets a reward from a distribution with density $f(x, \theta_i)$. Time is slotted, and players do not know the distributions (nor any statistics about them). The problem is to find a learning policy that minimizes the expected *regret* over some time horizon T . It was shown by Lai and Robbins [1] that there exists an index-type policy that achieves expected regret that grows asymptotically as $\log T$, and this is order-optimal, i.e., there exists no causal policy that can do better. This was generalized by Anantharam, *et al* [2] to the case of multiple plays, i.e., when the player can pick multiple arms at the same time. In [3], Agrawal proposed a sample mean based index policy which achieves $\log T$ regret asymptotically. Assuming that the rewards are coming from a distribution of bounded support, Auer, *et al* [4] proposed a much simpler sample mean based index policy, called UCB_1 , which achieves $\log T$ uniformly over time, not only asymptotically. Also, unlike the policy in [3], the index doesn't depend on the specific family of distributions that the rewards come from.

In [5], Anantharam, *et al* proposed a policy to the case where the arms are modelled as Markovian, not i.i.d. The rewards are assumed to come from a finite, irreducible and aperiodic Markov chain represented by a single parameter probability transition matrix. The state of each arm evolves according to an underlying transition probability matrix when the arm is played and remains frozen when passive. Such problems are called *rested Markovian bandit problems* (where *rested* refers to no state evolution until the arm is played). In [6], Tenkin and Liu extended the UCB_1 policy to the case of rested Markovian bandit problems. If some non-trivial bounds on the underlying Markov chains are known a priori, they showed that the policy achieves $\log T$ regret uniformly over time. Also, if no information about the underlying Markov chains is available, the policy can easily be modified to get a $\text{near-}O(\log T)$ regret asymptotically. The models in which the state of an arm continues to evolve even when it is not played are called *restless Markovian*

bandit problems. Restless models are considerably more difficult than the rested models and have been shown to be P-SPACE hard [7]. This is because the optimal policy no longer will be to “play the arm with the highest mean reward”. [8] employs a weaker notion of regret (*weak regret*) which compares the reward of a policy to that of a policy which always plays the the arm with the highest mean reward. They propose a policy which achieves $\log T$ (weak) regret uniformly over time if certain bounds on the underlying Markov model are known a priori and achieves a near- $O(\log T)$ (weak) regret asymptotically when no such knowledge is available. [9] proposes another simpler policy which achieves the same bounds for weak regret. [10] proposes a policy based on deterministic sequence of exploration and exploitation and achieves the same bounds for weak regret. In [11], the authors consider the notion of *strong regret* and propose a policy which achieves near- $\log T$ (strong) regret for some special cases of the restless model.

Recently, there is an increasing interest in multi-armed bandit models, partly because of opportunistic spectrum access problems. Consider a user who must choose between N wireless channels. Yet, it knows nothing about the channel statistics, i.e., has no idea of how good or bad the channels are, and what rate it may expect to get from each channel. The rates could be learnt by exploring various channels. Thus, these have been formulated as multi-armed bandit problems, and index-type policies have been proposed for choosing spectrum channels. In many scenarios, there are multiple users accessing the channels at the same time. Each of these users must be matched to a different channel. These have been formulated as a *combinatorial multi-armed bandit problem* [12] [13], and it was shown that an “index-matching” algorithm that at each instant determines a matching by solving a sum-index maximization problem achieves $O(\log T)$ regret uniformly over time, and this is indeed order-optimal.

In other settings, the users cannot coordinate, and the problem must be solved in a decentralized manner. Thus, settings where all channels (arms) are identical for all users with i.i.d. rewards have been considered, and index-type policies that can achieve coordination have been proposed that get $O(\log T)$ regret uniformly over time [14], [15], [16], [10]. A similar result for Markovian reward model with weak regret has been shown by [10], assuming some non-trivial bounds on the underlying Markov chains are known a priori. The regret scales only polynomially in the number

of users and channels. Surprisingly, the lack of coordination between the players asymptotically imposes no additional cost or regret.

In this paper, we consider the decentralized multi-armed bandit problem with distinct arms for each players. We consider both the i.i.d. reward model and the *rested* Markovian reward model. All players together must discover the best arms to play as a team. However, since they are all trying to learn at the same time, they may collide when two or more pick the same arm. We propose an index-type policy dUCB_4 based on a variation of the UCB_1 index. At its' heart is a distributed bipartite matching algorithm such as Bertsekas' auction algorithm [17]. This algorithm operates in rounds, and in each round prices for various arms are determined based on bid-values. This imposes communication (and computation) cost on the algorithm that must be accounted for. Nevertheless, we show that when certain non-trivial bounds on the model parameters are known a priori, the dUCB_4 algorithm that we introduce achieves (at most) $\text{near-}O(\log^2 T)$ growth non-asymptotically in expected regret. If no such information about the model parameters are available, dUCB_4 algorithm still achieves (at most) $\text{near-}O(\log^2 T)$ regret asymptotically. A lower bound, however, is not known at this point, and a work in progress.

The paper is organized as follows. In Section II, we present the model and problem formulation. In section III and IV we present some variations on single player MAB with i.i.d. rewards and Markovian rewards respectively. In section V, we introduce the decentralized MAB problem with i.i.d. rewards. We then extend the results to the decentralized cases with Markovian rewards in section VI. In section VII we present the distributed bipartite matching algorithm which is used in our main algorithm for decentralized MAB. In section VIII, we present some simulation results to numerically evaluate the performance of our algorithm.

II. MODEL AND PROBLEM FORMULATION

A. Arms with i.i.d. rewards

We consider an N -armed bandit with M players. In a wireless cognitive radio setting [18], each arm could correspond to a channel, and each player to a user who wants to use a channel. Time is slotted, and at each instant each player picks an arm. There is no dedicated control channel for coordination among the players. So, potentially more than one players can pick the

same arm at the same instant. We will regard that as a *collision*. Player i playing arm k at time t yields i.i.d. reward $S_{ik}(t)$ with univariate density function $f(s, \theta_{ik})$, where θ_{ik} is a parameter in the set Θ_{ik} . We will assume that the rewards are bounded, and without loss of generality lie in $[0, 1]$. Let $\mu_{i,k}$ denote the mean of $S_{ik}(t)$ w.r.t. the pdf $f(s, \theta_{ik})$. We assume that the parameter vector $\theta = (\theta_{ij}, 1 \leq i \leq M, 1 \leq j \leq N)$ is unknown to the players, i.e., the players have no information about the mean, the distributions or any other statistics about the rewards from various arms other than what they observe while playing. We also assume that each player can only observe the rewards that they get. When there is a collision, we will assume that all players that choose the arm on which there is a collision get zero reward. This could be relaxed where the players share the reward in some manner though the results do not change appreciably.

Let $X_{ij}(t)$ be the reward that player i gets from arm j at time t . Thus, if player i plays arm k at time t (and there is no collision), $X_{ik}(t) = S_{ik}(t)$, and $X_{ij}(t) = 0, j \neq k$. Denote the action of player i at time t by $a_i(t) \in \mathcal{A} := \{1, \dots, N\}$. Then, the *history* seen by player i at time t is $\mathcal{H}_i(t) = \{(a_i(1), X_{i,a_i(1)}(1)), \dots, (a_i(t), X_{i,a_i(t)}(t))\}$ with $\mathcal{H}_i(0) = \emptyset$. A *policy* $\alpha_i = (\alpha_i(t))_{t=1}^\infty$ for player i is a sequence of maps $\alpha_i(t) : \mathcal{H}_i(t) \rightarrow \mathcal{A}$ that specifies the arm to be played at time t given the history seen by the player. Let $\mathcal{P}(N)$ be the set of vectors such that

$$\mathcal{P}(N) := \{\mathbf{a} = (a_1, \dots, a_M) : a_i \in \mathcal{A}, a_i \neq a_j, \text{ for } i \neq j\}.$$

The players have a *team objective*: namely over a time horizon T , they want to maximize the expected sum of rewards $\mathbb{E}[\sum_{t=1}^T \sum_{i=1}^M X_{i,a_i(t)}(t)]$ over some time horizon T . If the parameters $\mu_{i,j}$ are known, this could easily be achieved by picking a bipartite matching

$$\mathbf{k}^{**} \in \arg \max_{\mathbf{k} \in \mathcal{P}(N)} \sum_{i=1}^M \mu_{i,k_i}, \quad (1)$$

i.e., the optimal bipartite matching with expected reward from each match. Note that this may not be unique. Since the expected rewards, $\mu_{i,j}$, are unknown, the players must pick learning policies that minimize the *expected regret*, defined for policies $\alpha = (\alpha_i, 1 \leq i \leq M)$ as

$$\mathcal{R}_\alpha(T) = T \sum_i \mu_{i,k_i^{**}} - \mathbb{E}_\alpha \left[\sum_{t=1}^T \sum_{i=1}^M X_{i,\alpha_i(t)}(t) \right]. \quad (2)$$

Our goal is to find a decentralized algorithm that players can use such that together they minimize the expected regret.

B. Arms with Markovian rewards

Here we follow the model formulation introduced in the previous subsection, with the exception that the rewards are now considered Markovian. The reward that player i gets from arm j (when there is no collision) X_{ij} , is modelled as an irreducible, aperiodic, reversible Markov chain on a finite state space $\mathcal{X}^{i,j}$ and represented by a transition probability matrix $P^{i,j} := (p_{x,x'}^{i,j} : x, x' \in \mathcal{X}^{i,j})$. We assume that rewards are bounded and strictly positive, and without loss of generality lie in $(0, 1]$. Let $\pi^{i,j} := (\pi_x^{i,j}, x \in \mathcal{X}^{i,j})$ be the stationary distribution of the Markov chain $P^{i,j}$. The mean reward from arm j for player i is defined as $\mu_{i,j} := \sum_{x \in \mathcal{X}^{i,j}} x \pi_x^{i,j}$. Note that the Markov chain represented by $P^{i,j}$ makes a state transition only when player i plays arm j . Otherwise it remains *rested*.

We note that although we use the ‘big O ’ notation to emphasis the regret order, unless otherwise noted results are non-asymptotic.

III. SOME VARIATIONS ON SINGLE PLAYER MULTI-ARMED BANDIT WITH I.I.D. REWARDS

We first present some variations on the single player non-Bayesian multi-armed bandit model. These will prove useful later for the multi-player problem though they should also be of independent interest.

A. UCB₁ with index recomputation every L slots

Consider the classical single player non-Bayesian N -armed bandit problem. At each time t , the player picks a particular arm, say j , and gets a random reward $X_j(t)$. The rewards $X_j(t), 1 \leq t \leq T$ are independent and identically distributed according to some unknown probability measure with an unknown expectation μ_j . Without loss of generality, assume that $\mu_1 > \mu_i > \mu_N$, for $i = 2, \dots, N-1$. Let $n_j(t)$ denote the number of times arm j has been played by time t . Denote $\Delta_j := \mu_1 - \mu_j$, $\Delta_{\min} := \min_{j,j \neq 1} \Delta_j$ and $\Delta_{\max} := \max_j \Delta_j$. The regret for any policy α is

$$\mathcal{R}_\alpha(T) := \mu_1 T - \sum_{j=1}^N \mu_j \mathbb{E}_\alpha[n_j(T)]. \quad (3)$$

UCB₁ index [4] is defined as

$$g_j(t) := \bar{X}_j(t) + \sqrt{\frac{2 \log(t)}{n_j(t)}}, \quad (4)$$

where $\bar{X}_j(t)$ is the average reward obtained by playing arm j by time t . It is defined as $\bar{X}_j(t) = \sum_{m=1}^t r_j(m)/n_j(t)$, where $r_j(m)$ is the reward obtained from arm j at time m . If the arm j is played at time t then $r_j(m) = X_j(m)$ and otherwise $r_j(t) = 0$. Now, an index-based policy called UCB₁ [4] is to pick the arm that has the highest index at each instant. It can be shown that this algorithm achieves regret that grows logarithmically in T non-asymptotically.

An easy variation of the above algorithm which will be useful in our analysis of subsequent algorithms is the following. Suppose the index is re-computed only once every L slots. In that case, it is easy to establish the following.

Theorem 1. *Under the UCB₁ algorithm with recomputation of the index once every L slots, the expected regret by time T is given by*

$$\mathcal{R}_{\text{UCB}_1}(T) \leq \sum_{j>1}^N \frac{8L \log T}{\Delta_j} + L \left(1 + \frac{\pi^2}{3}\right) \sum_{j>1}^N \Delta_j. \quad (5)$$

The proof follows [4] and taking into account the fact that every time a suboptimal arm is selected, it is played for the next L time slots. We omit it due to space consideration.

B. UCB₄ Algorithm when index computation is costly

Often, learning algorithms pay a penalty or cost for computation. This is particularly the case when the algorithms must solve combinatorial optimization problems that are NP-hard. Such costs also arise in decentralized settings wherein algorithms pay a communication cost for coordination between the decentralized players. This is indeed the case, as we shall see later when we present an algorithm to solve the decentralized multi-armed bandit problem. Here, however, we will just consider an “abstract” communication or computation cost. The problem we formulate below can be solved with better regret bounds than what we present. At this time though we are unable to design algorithms with better regret bounds, that also help in decentralization.

Consider a computation cost every time the index is recomputed. Let the cost be C units. Let $m(t)$ denote the number of times the index is computed by time t . Then, under policy α the expected regret is now given by

$$\tilde{\mathcal{R}}_\alpha(T) := \mu_1 T - \sum_{j=1}^N \mu_j \mathbb{E}_\alpha[n_j(T)] + C \mathbb{E}_\alpha[m(T)]. \quad (6)$$

It is easy to argue that the UCB_1 algorithm will give a regret $\Omega(T)$ for this problem. We present an alternative algorithm called UCB_4 algorithm, that gives sub-linear regret. Define the UCB_4 index

$$g_j(t) := \bar{X}_j(t) + \sqrt{\frac{3 \log(t)}{n_j(t)}}. \quad (7)$$

We define an arm $j^*(t)$ to be the *best arm* if $j^*(t) \in \arg \max_{1 \leq i \leq N} g_i(t)$.

Algorithm 1 : UCB_4

```

1: Initialization: Select each arm  $j$  once for  $t \leq N$ . Update the  $\text{UCB}_4$  indices. Set  $\eta = 1$ .
2: while ( $t \leq T$ ) do
3:   if ( $\eta = 2^p$  for some  $p = 0, 1, 2, \dots$ ) then
4:     Update the index vector  $g(t)$ ;
5:     Compute the best arm  $j^*(t)$ ;
6:     if ( $j^*(t) \neq j^*(t-1)$ ) then
7:       Reset  $\eta = 1$ ;
8:     end if
9:   else
10:     $j^*(t) = j^*(t-1)$ ;
11:  end if
12:  Play arm  $j^*(t)$ ;
13:  Increment counter  $\eta = \eta + 1$ ;  $t = t + 1$ ;
14: end while

```

We will use the following concentration inequality.

Fact 1: Chernoff-Hoeffding inequality [19]

Let X_1, \dots, X_t be random variables with a common range such that $\mathbb{E}[X_t | X_1, \dots, X_{t-1}] = \mu$.

Let $S_t = \sum_{i=1}^t X_i$. Then for all $a \geq 0$,

$$\mathbb{P}(S_t \geq t\mu + a) \leq e^{-2a^2/t}, \quad \text{and} \quad \mathbb{P}(S_t \leq t\mu - a) \leq e^{-2a^2/t}. \quad (8)$$

Theorem 2. *The expected regret for the single player multi-armed bandit problem with per*

computation cost C using the UCB_4 algorithm is given by

$$\tilde{\mathcal{R}}_{\text{UCB}_4}(T) \leq (\Delta_{\max} + C(1 + \log T)) \cdot \left(\sum_{j>1}^N \frac{12 \log T}{\Delta_j^2} + 2N \right).$$

Thus, $\tilde{\mathcal{R}}_{\text{UCB}_4}(T) = O(\log^2 T)$.

Proof: We prove this in two steps. First, we compute the expected number of times a suboptimal arm is played and then the expected number of times we recompute the index.

Consider any suboptimal arm $j > 1$. Denote $c_{t,s} = \sqrt{3 \log t / s}$ and the indicator function of the event A by $I\{A\}$. let $\tau_{j,m}$ be the time at which the player makes the m th transition to the arm j from another arm and $\tau'_{j,m}$ be the time at which the player makes the m th transition from the arm j to another arm. Let $\tilde{\tau}'_{j,m} = \min\{\tau'_{j,m}, T\}$. Then,

$$\begin{aligned} n_j(T) &\leq 1 + \sum_{m=1}^T (\tilde{\tau}'_{j,m} - \tau_{j,m}) I\{\text{Arm } j \text{ is picked at time } \tau_{j,m}, \tau_{j,m} \leq T\} \\ &\leq 1 + \sum_{m=1}^T (\tilde{\tau}'_{j,m} - \tau_{j,m}) I\{g_j(\tau_{j,m} - 1) \geq g_1(\tau_{j,m} - 1), \tau_{j,m} \leq T\} \\ &\leq l + \sum_{m=1}^T (\tilde{\tau}'_{j,m} - \tau_{j,m}) I\{g_j(\tau_{j,m} - 1) \geq g_1(\tau_{j,m} - 1), \tau_{j,m} \leq T, n_j(\tau_{j,m} - 1) \geq l\} \\ &\stackrel{(a)}{\leq} l + \sum_{m=1}^T \sum_{p=0}^{\infty} 2^p I\{g_j(\tau_{j,m} + 2^p - 2) \geq g_1(\tau_{j,m} + 2^p - 2), \tau_{j,m} + 2^p \leq T, n_j(\tau_{j,m} - 1) \geq l\} \\ &\stackrel{(b)}{\leq} l + \sum_{m=2}^T \sum_{p=0}^{\infty} 2^p I\{g_j(m + 2^p - 2) \geq g_1(m + 2^p - 2), m + 2^p \leq T, n_j(m - 1) \geq l\} \\ &\leq l + \sum_{m=1}^T \sum_{p \geq 0, m+2^p \leq T} 2^p I\{\bar{X}_j(m + 2^p - 1) + c_{m+2^p-1, n_j(m+2^p-1)} \geq \\ &\quad \bar{X}_1(m + 2^p - 1) + c_{m+2^p-1, n_1(m+2^p-1)}, n_j(m - 1) \geq l\} \tag{9} \\ &\leq l + \sum_{m=1}^T \sum_{p \geq 0, m+2^p \leq T} 2^p I\{\max_{l \leq s_j < m+2^p} \bar{X}_j(m + 2^p - 1) + c_{m+2^p-1, s_j} \geq \\ &\quad \min_{1 \leq s_1 < m+2^p} \bar{X}_1(m + 2^p - 1) + c_{m+2^p-1, s_1}\} \\ &\leq l + \sum_{m=1}^{\infty} \sum_{p \geq 0, m+2^p \leq T} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=l}^{m+2^p} I\{\bar{X}_j(m + 2^p) + c_{m+2^p, s_j} \geq \bar{X}_1(m + 2^p) + c_{m+2^p, s_1}\} \tag{10} \end{aligned}$$

In Algorithm 1 (UCB_4), if an arm is for the p th time consecutively (without switching to any

other arms in between), it is be played for the next 2^p slots. Inequality (a) uses this fact. In the inequality (b), we replace $\tau_{j,m}$ by m which is clearly an upper bound. Now, observe that the event $\{\bar{X}_j(m + 2^p) + c_{m+2^p,s_j} \geq \bar{X}_1(m + 2^p) + c_{m+2^p,s_1}\}$ implies at least one of the following events,

$$\begin{aligned} A &:= \{\bar{X}_1(m + 2^p) \leq \mu_1 - c_{m+2^p,s_1}\}, \quad B := \{\bar{X}_j(m + 2^p) \geq \mu_j + c_{m+2^p,s_j}\}, \\ \text{or } C &:= \{\mu_1 < \mu_j + 2c_{m+2^p,s_j}\}. \end{aligned} \quad (11)$$

Now, using the Chernoff-Hoeffding bound, we get

$$\mathbb{P}(\bar{X}_1(m + 2^p) \leq \mu_1 - c_{m+2^p,s_1}) \leq (m + 2^p)^{-6}, \quad \mathbb{P}(\bar{X}_j(m + 2^p) \geq \mu_j + c_{m+2^p,s_j}) \leq (m + 2^p)^{-6}.$$

For $l = \left\lceil \frac{12 \log T}{\Delta_j^2} \right\rceil$, the last event in (11) is false. In fact, $\mu_1 - \mu_j - 2c_{m+2^p,s_j}$

$$= \mu_1 - \mu_j - 2\sqrt{3 \log(m + 2^p)/s_j} \geq \mu_1 - \mu_j - \Delta_j = 0, \quad \text{for } s_j \geq \lceil 12 \log T / \Delta_j^2 \rceil.$$

$$\begin{aligned} \text{So, we get, } \mathbb{E}[n_j(T)] &\leq \lceil 12 \log T / \Delta_j^2 \rceil + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=1}^{m+2^p} 2(m + 2^p)^{-6} \\ &\leq \lceil 12 \log T / \Delta_j^2 \rceil + 2 \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m + 2^p)^{-4} \leq \frac{12 \log T}{\Delta_j^2} + 2. \end{aligned} \quad (12)$$

Next, we upper-bound the expectation of $m(T)$, the number of index computations performed by time T . We can write $m(T) = m_1(T) + m_2(T)$, where $m_1(T)$ is the number of index updates that result in an optimal allocation, and $m_2(T)$ is the number of index updates that result in a suboptimal allocation. Clearly, the number of updates resulting in a suboptimal allocation is less than the number of times a suboptimal arm is played. Thus,

$$\mathbb{E}[m_2(T)] \leq \sum_{j>1}^N \mathbb{E}[n_j(T)]. \quad (13)$$

To bound $\mathbb{E}[m_1(T)]$, let τ_l be the time at which the player makes the l th transition to an optimal arm from a suboptimal arm and τ'_l be the time at which the player makes the l th transition from an optimal arm to a suboptimal arm. Then, $m_1(T) \leq \sum_{l=1}^{n_{sub}(T)} \log |\tau_l - \tau'_l|$, where $n_{sub}(T)$ is the total number of such transitions by time T . Clearly, $n_{sub}(T)$ is upper-bounded by the total

number of times the player picks a sub-optimal arm. Also, $\log |\tau_l - \tau'_l| \leq \log T$. So,

$$\mathbb{E}[m_1(T)] \leq \sum_{j>1}^N \mathbb{E}[n_j(T)] \cdot \log T. \quad (14)$$

Thus, from bounds (13) and (14), we get

$$\mathbb{E}[m(T)] \leq \sum_{j>1}^N \mathbb{E}[n_j(T)] \cdot (1 + \log T). \quad (15)$$

Now, using equation (6), the expected regret is

$$\begin{aligned} \tilde{\mathcal{R}}_{\text{UCB}_4}(T) &= \sum_{j>1}^N \mathbb{E}[n_j(T)] \cdot \Delta_j + C\mathbb{E}[m(T)] \leq \Delta_{\max} \sum_{j>1}^N \mathbb{E}[n_j(T)] + C\mathbb{E}[m(T)] \\ &\leq (\Delta_{\max} + C(1 + \log T)) \sum_{j>1}^N \mathbb{E}[n_j(T)]. \end{aligned}$$

by using (15). Now, by bound (12), we get the desired bound on the expected regret. ■

Remarks. 1. It is easy to show that the lower bound for the single player MAB problem with computation costs is $\Omega(\log T)$. This can be achieved by the UCB_2 algorithm [4]. To see this, note that the number of times the player selects a suboptimal arm when using UCB_2 is $O(\log T)$. Since $\mathbb{E}[n_j(T)] = O(\log T)$, we get $\mathbb{E}[\sum_{j>1}^N n_j(T)] = O(\log T)$, and also $\mathbb{E}[m_2(T)] = O(\log T)$. Now, since the epochs are not getting reset after every switch and are exponentially spaced, the number of updates that result in the optimal allocation, $m_1(T) \leq \log T$. These together yield

$$\tilde{\mathcal{R}}_{\text{UCB}_2}(T) \leq \sum_{j>1}^N \mathbb{E}[n_j(T)] \cdot \Delta_j + C\mathbb{E}[m(T)] = O(\log T).$$

2. Variations of the UCB_2 algorithm that use a deterministic schedule can also be used [20]. But it is unknown at this time if these can be used in solving the decentralized MAB problem that we introduce in the next section. This is the main reason for introducing the UCB_4 algorithm.

C. Algorithms with finite precision indices

Often, there might be a cost to compute the indices to a particular precision. In that case, indices may be known upto some ϵ precision, and it may not be possible to tell which of two indices is greater if they are within ϵ of each other. The question then is how is the performance

of various index-based policies such as UCB_1 , UCB_4 , etc. affected if there are limits on index resolution, and only an arm with an ϵ -highest index can be picked. We first show that if Δ_{\min} is known, we can fix a precision $0 < \epsilon < \Delta_{\min}$, so that UCB_4 algorithm will achieve order log-squared regret growth with T . If Δ_{\min} is not known, we can pick a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \rightarrow 0$, as $t \rightarrow \infty$. Denote the cost of computation for ϵ -precision be $C(\epsilon)$. We assume that $C(\epsilon) \rightarrow \infty$ monotonically as $\epsilon \rightarrow 0$.

Theorem 3. (i) If Δ_{\min} is known, choose an $0 < \epsilon < \Delta_{\min}$. Then, the expected regret of the UCB_4 algorithm with ϵ -precise computations is given by

$$\tilde{\mathcal{R}}_{\text{UCB}_4}(T) \leq (\Delta_{\max} + C(\epsilon)(1 + \log T)) \cdot \left(\sum_{j>1}^N \frac{12 \log T}{(\Delta_j - \epsilon)^2} + 2N \right).$$

Thus, $\tilde{\mathcal{R}}_{\text{UCB}_4}(T) = O(\log^2 T)$.

(ii) If Δ_{\min} is unknown, denote $\epsilon_{\min} = \Delta_{\min}/2$ and choose a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \rightarrow 0$ as $t \rightarrow \infty$. Then, there exists a $t_0 > 0$ such that for all $T > t_0$,

$$\tilde{\mathcal{R}}_{\text{UCB}_4}(T) \leq (\Delta_{\max} + C(\epsilon_{\min})) t_0 + (\Delta_{\max} + C(\epsilon_T)(1 + \log T)) \cdot \left(\sum_{j>1}^N \frac{12 \log T}{(\Delta_j - \epsilon_{\min})^2} + 2N \right)$$

where t_0 is the smallest t such that $\epsilon_{t_0} < \epsilon_{\min}$. Thus by choosing an arbitrarily slowly increasing sequence $\{\epsilon_t\}$, we can make the regret arbitrarily close to $O(\log^2 T)$ asymptotically.

Proof: (i) The proof is only a slight modification of the proof given in Theorem 2. Due to the ϵ precision, the player will pick a suboptimal arm if the event $\{\bar{X}_j(m + 2^p) + c_{m+2^p, s_j} + \epsilon \geq \bar{X}_1(m + 2^p) + c_{m+2^p, s_1}\}$ occurs. Thus equation (9) becomes, $n_j(T)$

$$\leq l + \sum_{m=1}^{\infty} \sum_{p \geq 0, m+2^p \leq T} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=l}^{m+2^p} I\{\bar{X}_j(m + 2^p) + c_{m+2^p, s_j} + \epsilon \geq \bar{X}_1(m + 2^p) + c_{m+2^p, s_1}\}.$$

Now, the event $\{\bar{X}_j(m + 2^p) + c_{m+2^p, s_j} + \epsilon \geq \bar{X}_1(m + 2^p) + c_{m+2^p, s_1}\}$ implies that at least one of the following events must occur:

$$\begin{aligned} A &:= \{\bar{X}_1(m + 2^p) \leq \mu_1 - c_{m+2^p, s_1}\}, & B &:= \{\bar{X}_j(m + 2^p) \geq \mu_j + \epsilon + c_{m+2^p, s_j}\}, \\ C &:= \{\mu_1 < \mu_j + \epsilon + 2c_{m+2^p, s_j}\}, & \text{or } D &:= \{\mu_1 < \mu_j + \epsilon\}. \end{aligned} \tag{16}$$

Since $\{\overline{X}_j(m + 2^p) \geq \mu_j + \epsilon + c_{m+2^p, s_j}\} \subseteq \{\overline{X}_j(m + 2^p) \geq \mu_j + c_{m+2^p, s_j}\}$, we have

$$\mathbb{P}(\{\overline{X}_j(m + 2^p) \geq \mu_j + \epsilon + c_{m+2^p, s_j}\}) \leq \mathbb{P}(\{\overline{X}_j(m + 2^p) \geq \mu_j + c_{m+2^p, s_j}\}).$$

Also, for $l = \lceil 12 \log T / (\Delta_j - \epsilon)^2 \rceil$, the event C cannot happen. In fact, $\mu_1 - \mu_j - \epsilon - 2c_{l+2^p, s_j} = \mu_1 - \mu_j - \epsilon - 2\sqrt{\frac{3 \log(l+2^p)}{s_j}} \geq \mu_1 - \mu_j - \epsilon - (\Delta_j - \epsilon) = 0$, for $s_j \geq \lceil 12 \log T / (\Delta_j - \epsilon)^2 \rceil$. If $\epsilon < \Delta_{\min}$, the last event (D) in equation (16) is also not true. Thus, for $0 < \epsilon < \Delta_{\min}$, we get

$$\mathbb{E}[n_j(T)] \leq \frac{12 \log(n)}{(\Delta_j - \epsilon)^2} + 2. \quad (17)$$

The rest of the proof is the same as in Theorem 2. Now, if Δ_{\min} is known, we can choose $0 < \epsilon < \Delta_{\min}$ and by Theorem 2 and bound (17), we get the desired result.

(ii) If Δ_{\min} is unknown, we can choose a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \rightarrow 0$ as $t \rightarrow \infty$. Thus, there exists a t_0 such that for $t > t_0$, $\epsilon_t < \epsilon_{\min}$. We may get a linear regret upto time t_0 but after that the analysis follows that in the proof of Theorem 2, and regret grows only sub-linearly. Since $C(\cdot)$ is monotone, $C(\epsilon_T) > C(\epsilon_t)$ for all $t < T$. The last part can now be trivially established using the obtained bound on the expected regret. ■

IV. SINGLE PLAYER MULTI-ARMED BANDIT WITH MARKOVIAN REWARDS

Now, we consider the scenario where the rewards obtained from an arm are not i.i.d. but come from a Markov chain. Reward from each arm is modelled as an irreducible, aperiodic, reversible Markov chain on a finite state space \mathcal{X}^i and represented by a transition probability matrix $P^i := (p_{x, x'}^i : x, x' \in \mathcal{X}^i)$. Assume that the reward space $\mathcal{X}^i \subseteq (0, 1]$. Let $X_i(1), X_i(2), \dots$ denote the successive rewards from arm i . All arms are mutually independent. Let $\pi^i := (\pi_x^i, x \in \mathcal{X}^i)$ be the stationary distribution of the Markov chain P^i . Since the Markov chains are ergodic under these assumptions, the mean reward from arm i is given by $\mu_i := \sum_{x \in \mathcal{X}^i} x \pi_x^i$. Without loss of generality, assume that $\mu_1 > \mu_i > \mu_N$, for $i = 2, \dots, N-1$. As before, $n_j(t)$ denotes the number of times arm j has been played by time t . Denote $\Delta_j := \mu_1 - \mu_j$, $\Delta_{\min} := \min_{j, j \neq 1} \Delta_j$ and $\Delta_{\max} := \max_j \Delta_j$. Denote $\pi_{\min} := \min_{1 \leq i \leq N, x \in \mathcal{X}^i} \pi_x^i$, $x_{\max} := \max_{1 \leq i \leq N, x \in \mathcal{X}^i} x$ and $x_{\min} := \min_{1 \leq i \leq N, x \in \mathcal{X}^i} x$. Let $\hat{\pi}_x^i := \max\{\pi_x^i, 1 - \pi_x^i\}$ and $\hat{\pi}_{\max}^i := \max_{1 \leq i \leq N, x \in \mathcal{X}^i} \hat{\pi}_x^i$. Let $|\mathcal{X}^i|$ denote

the cardinality of the state space \mathcal{X}^i , $|\mathcal{X}|_{max} := \max_{1 \leq i \leq N} |\mathcal{X}^i|$. Let ρ^i be the eigenvalue gap, $1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the matrix P^{i2} . Denote $\rho_{max} := \max_{1 \leq i \leq N} \rho^i$ and $\rho_{min} := \min_{1 \leq i \leq N} \rho^i$, where ρ^i is the eigenvalue gap of the i th arm.

The total reward obtained by the time T is then given by $S_T = \sum_{j=1}^N \sum_{s=1}^{n_j(T)} X_j(s)$. The regret for any policy α is defined as

$$\tilde{\mathcal{R}}_{M,\alpha}(T) := \mu_1 T - \mathbb{E}_\alpha \sum_{j=1}^N \sum_{s=1}^{n_j(T)} X_j(s) + C \mathbb{E}_\alpha[m(T)] \quad (18)$$

where C is the cost per computation and $m(T)$ is the number of times the index is computed by time T , as described in section III. Define the index

$$g_j(t) := \bar{X}_j(t) + \sqrt{\frac{\kappa \log(t)}{n_j(t)}}, \quad (19)$$

where $\bar{X}_j(t)$ is the average reward obtained by playing arm j by time t , as defined in the previous section. κ can be any constant satisfying $\kappa > 168|\mathcal{X}|_{max}^2/\rho_{min}$.

We introduce one more notation here. If \mathcal{F} and \mathcal{G} are two σ -algebras, then $\mathcal{F} \vee \mathcal{G}$ denotes the smallest σ -algebra containing \mathcal{F} and \mathcal{G} . Similarly, if $\{\mathcal{F}_t, t = 1, 2, \dots\}$ is a collection of σ -algebras, then $\vee_{t \geq 1} \mathcal{F}_t$ denotes the smallest σ -algebra containing $\mathcal{F}_1, \mathcal{F}_2, \dots$.

The following can be derived easily from Lemma 4 [5], reproduced in the appendix.

Lemma 1. *If the reward of each arm is given by a Markov chain satisfying the hypothesis of Lemma 4, then under any policy α we have*

$$\tilde{\mathcal{R}}_{M,\alpha}(T) \leq \sum_{j=2}^N \Delta_j \mathbb{E}_\alpha[n_j(T)] + K_{\mathcal{X},P} + C \mathbb{E}_\alpha[m(T)] \quad (20)$$

where $K_{\mathcal{X},P} = \sum_{j=1}^N \sum_{x \in \mathcal{X}^j} x / \pi_{min}^j$ and $\pi_{min}^j = \min_{x \in \mathcal{X}^j} \pi_x^j$

Proof: Let $X_j(1), X_j(2), \dots$ denote the successive rewards from arm j . Let \mathcal{F}_t^j denotes the σ -algebra generated by $(X_j(1), \dots, X_j(t))$. Let $\mathcal{F}^j = \vee_{t \geq 1} \mathcal{F}_t^j$ and $\mathcal{G}^j = \vee_{i \neq j} \mathcal{F}^i$. Since arms are independent, \mathcal{G}^j is independent of \mathcal{F}^j . Clearly, $n_j(T)$ is a stopping time with respect to $\mathcal{G}^j \vee \mathcal{F}_T^j$. The total reward is $S_T = \sum_{j=1}^N \sum_{s=1}^{n_j(T)} X_j(s) = \sum_{j=1}^N \sum_{x \in \mathcal{X}^j} x N(x, n_j(T))$ where

$N(x, n_j(T)) := \sum_{t=1}^{n_j(T)} I\{X_j(t) = x\}$. Taking the expectation and using the Lemma 4, we have $|\mathbb{E}[S_T] - \sum_{j=1}^N \sum_{x \in \mathcal{X}^j} x \pi_x^j \mathbb{E}[n_j(T)]| \leq \sum_{j=1}^N \sum_{x \in \mathcal{X}^j} x / \pi_{min}^j$, which implies $|\mathbb{E}[S_T] - \sum_{j=1}^N \mu_j \mathbb{E}[n_j(T)]| \leq K_{\mathcal{X},P}$, where $K_{\mathcal{X},P} = \sum_{j=1}^N \sum_{x \in \mathcal{X}^j} x / \pi_{min}^j$. Since regret $\tilde{\mathcal{R}}_{M,\alpha}(T) = \mu_1 T - \mathbb{E}_\alpha \sum_{j=1}^N \sum_{t=1}^{n_j(T)} X_j(t) + C \mathbb{E}_\alpha[m(T)]$ (c.f. equation (18)), we get

$$|\tilde{\mathcal{R}}_{M,\alpha}(T) - \left(\mu_1 T - \sum_{j=1}^N \mu_j \mathbb{E}[n_j(T)] + C \mathbb{E}_\alpha[m(T)] \right)| \leq K_{\mathcal{X},P}.$$

■

We will use a concentration inequality for Markov chains (Lemma 5, from [21]), reproduced in the appendix.

Theorem 4. (i) If $|\mathcal{X}|_{max}$ and ρ_{min} are known, choose $\kappa > 168|\mathcal{X}|_{max}^2/\rho_{min}$. Then, the expected regret using the UCB₄ algorithm with the index defined as in (19) for the single player multi-armed bandit problem with Markovian rewards and per computation cost C is given by

$$\tilde{\mathcal{R}}_{M,UCB_4}(T) \leq (\Delta_{max} + C(1 + \log T)) \cdot \left(\sum_{j>1}^N \frac{4\kappa \log T}{\Delta_j^2} + N(2D + 1) \right) + K_{\mathcal{X},P}$$

where $D = \frac{|\mathcal{X}|_{max}}{\pi_{min}}$. Thus, $\tilde{\mathcal{R}}_{M,UCB_4}(T) = O(\log^2 T)$.

(ii) If $|\mathcal{X}|_{max}$ and ρ_{min} are not known, choose a positive monotone sequence $\{\kappa_t\}$ such that $\kappa_t \rightarrow \infty$ as $t \rightarrow \infty$ and $\kappa_t \leq t$. Then, $\tilde{\mathcal{R}}_{M,UCB_4}(T) = O(\kappa_T \log^2 T)$. Thus, by choosing an arbitrarily slowly increasing sequence $\{\kappa_t\}$ we can make the regret arbitrarily close to $\log^2 T$.

Proof: (i) Consider any suboptimal arm $j > 1$. Denote $c_{t,s} = \sqrt{\kappa \log t/s}$. As in the proof of Theorem 2, we start by bounding $n_j(T)$. The initial steps are the same as in the proof of Theorem 2. So, we skip those steps and start from the inequality (9) there.

$$n_j(T) \leq l + \sum_{m=1}^{\infty} \sum_{p \geq 0, m+2^p \leq T} \sum_{s_1=1}^{m+2^p} \sum_{s_j=l}^{m+2^p} 2^p I\{\bar{X}_j(m+2^p) + c_{m+2^p,s_j} \geq \bar{X}_1(m+2^p) + c_{m+2^p,s_1}\}.$$

The event $\{\bar{X}_j(m+2^p) + c_{m+2^p,s_j} \geq \bar{X}_1(m+2^p) + c_{m+2^p,s_1}\}$ is true only if at least one of the events shown in display (11) are true. We note that, for any initial distribution λ^j for arm j ,

$$N_{\lambda^j} = \left\| \left(\frac{\lambda_x^j}{\pi_x^j}, x \in \mathcal{X}^j \right) \right\|_2 \leq \sum_{x \in \mathcal{X}^j} \left\| \left(\frac{\lambda_x^j}{\pi_x^j} \right) \right\|_2 \leq \frac{1}{\pi_{\min}}. \quad (21)$$

Also, $x_{\max} \leq 1$. Let $n_x^j(s_j)$ be the number of times the state x is observed when arm j is pulled s_j times. Then, the probability of the first event in (11),

$$\begin{aligned} \mathbb{P}(\overline{X}_j(m+2^p) \geq \mu_j + c_{m+2^p, s_j}) &= \mathbb{P} \left(\sum_{x \in \mathcal{X}^j} x n_x^j(s_j) \geq s_j \sum_{x \in \mathcal{X}^j} x \pi_x^j + s_j c_{m+2^p, s_j} \right) = \mathbb{P} \left(\sum_{x \in \mathcal{X}^j} (n_x^j(s_j) - s_j \pi_x^j) \geq s_j c_{m+2^p, s_j} / x \right) \\ &\stackrel{(a)}{\leq} \sum_{x \in \mathcal{X}^j} \mathbb{P} \left(n_x^j(s_j) - s_j \pi_x^j \geq \frac{s_j c_{m+2^p}}{x |\mathcal{X}^j|} \right) = \sum_{x \in \mathcal{X}^j} \mathbb{P} \left(\frac{\sum_{t=1}^{s_j} I\{X_j(t) = x\} - s_j \pi_x^j}{s_j \hat{\pi}_x^j} \geq \frac{c_{m+2^p, s_j}}{x |\mathcal{X}^j| \hat{\pi}_x^j} \right) \\ &\stackrel{(b)}{\leq} \sum_{x \in \mathcal{X}^j} N_{\lambda^j}(m+2^p)^{-\kappa \rho^j / 28 x^2 |\mathcal{X}^j|^2 (\hat{\pi}_x^j)^2} \stackrel{(c)}{\leq} \frac{|\mathcal{X}|_{\max}}{\pi_{\min}} (m+2^p)^{-\kappa \rho_{\min} / 28 |\mathcal{X}|_{\max}^2}. \end{aligned}$$

The inequality (a) follows after some simple algebra, which we skip due to space limitations.

The inequality (b) follows by defining the function $f(X_j(t)) = (I\{X_j(t) = x\} - \pi_x^j) / \hat{\pi}_x^j$ and using the Lemma 5. For inequality (c) we used the facts that $N_{\lambda^j} \leq 1/\pi_{\min}$, $x_{\max} \leq 1$ and $\hat{\pi}_{\max} \leq 1$. Thus,

$$\mathbb{P}(\overline{X}_j(m+2^p) \geq \mu_j + c_{m+2^p, s_j}) \leq D(m+2^p)^{-\kappa \rho_{\min} / 28 |\mathcal{X}|_{\max}^2} \quad (22)$$

where $D = \frac{|\mathcal{X}|_{\max}}{\pi_{\min}}$. Similarly we can get,

$$\mathbb{P}(\overline{X}_1(m+2^p) \leq \mu_1 - c_{m+2^p, s_1}) \leq D(m+2^p)^{-\kappa \rho_{\min} / 28 |\mathcal{X}|_{\max}^2} \quad (23)$$

For $l = \lceil 4\kappa \log T / \Delta_j^2 \rceil$, the last event in (11) is false. In fact, $\mu_1 - \mu_j - 2c_{m+2^p, s_j}$

$$= \mu_1 - \mu_j - 2\sqrt{\kappa \log(m+2^p) / s_j} \geq \mu_1 - \mu_j - \Delta_j = 0, \text{ for } s_j \geq \lceil 4\kappa \log T / \Delta_j^2 \rceil. \text{ Thus,}$$

$$\mathbb{E}[n_j(T)] \leq \left\lceil \frac{4\kappa \log T}{\Delta_j^2} \right\rceil + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=1}^{m+2^p} 2D(m+2^p)^{-\frac{\kappa \rho_{\min}}{28 |\mathcal{X}|_{\max}^2}}$$

$$= \left\lceil \frac{4\kappa \log T}{\Delta_j^2} \right\rceil + 2D \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m + 2^p)^{-\frac{\kappa \rho_{\min} - 56|\mathcal{X}|_{\max}^2}{28|\mathcal{X}|_{\max}^2}}. \quad (24)$$

When $\kappa > 168|\mathcal{X}|_{\max}^2/\rho_{\min}$, the above summation converges to a value less than 1 and we get

$$\mathbb{E}[n_j(T)] \leq \frac{4\kappa \log T}{\Delta_j^2} + (2D + 1). \quad (25)$$

Now, from the proof of Theorem 2 (equation (15)),

$$\mathbb{E}[m(T)] \leq \sum_{j>1}^N \mathbb{E}[n_j(T)] \cdot (1 + \log T). \quad (26)$$

Now, using inequality (20), the expected regret $\tilde{\mathcal{R}}_{M, \text{UCB}_4}(T) =$

$$\begin{aligned} &= \sum_{j>1}^N \mathbb{E}[n_j(T)] \cdot \Delta_j + C\mathbb{E}[m(T)] + K_{\mathcal{X},P} \leq \Delta_{\max} \sum_{j>1}^N \mathbb{E}[n_j(T)] + C\mathbb{E}[m(T)] + K_{\mathcal{X},P} \\ &\leq (\Delta_{\max} + C(1 + \log T)) \sum_{j>1}^N \mathbb{E}[n_j(T)] + K_{\mathcal{X},P}. \end{aligned}$$

by using (26). Now, by bound (25), we get the desired bound on the expected regret.

(ii) Replacing κ with κ_t , equation (24) becomes

$$\mathbb{E}[n_j(T)] \leq \left\lceil \frac{4\kappa_T \log T}{\Delta_j^2} \right\rceil + 2D \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m + 2^p)^{-\frac{\kappa_{m+2^p} \rho_{\min} - 56|\mathcal{X}|_{\max}^2}{28|\mathcal{X}|_{\max}^2}}$$

Since, $\kappa_t \rightarrow \infty$ as $t \rightarrow \infty$, the exponent $-\frac{\kappa_{m+2^p} \rho_{\min} - 56|\mathcal{X}|_{\max}^2}{28|\mathcal{X}|_{\max}^2}$ becomes smaller than -4 for sufficiently large m and p , and the above summation converges, yielding the desired result. ■

We note that we have used the results in [22] in the above proof. We note that the results for Markovian reward just presented extend easily even with finite precision indices. As before, suppose the cost of computation for ϵ -precision is $C(\epsilon)$. We assume that $C(\epsilon) \rightarrow \infty$ monotonically as $\epsilon \rightarrow 0$. We formally state the following result, which we will use in section VI.

Theorem 5. (i) If Δ_{\min} , $|\mathcal{X}|_{\max}$ and ρ_{\min} are known, choose an $0 < \epsilon < \Delta_{\min}$, and a $\kappa > 168|\mathcal{X}|_{\max}^2/\rho_{\min}$. Then, the expected regret using the UCB_4 algorithm with the index defined as in (19) for the single player multi-armed bandit problem with Markovian rewards with ϵ -precise

computations is given by

$$\tilde{\mathcal{R}}_{M,\text{UCB}_4}(T) \leq (\Delta_{\max} + C(\epsilon)(1 + \log T)) \cdot \left(\sum_{j>1}^N \frac{4\kappa \log T}{(\Delta_j - \epsilon)^2} + N(2D + 1) \right).$$

where $D = \frac{|\mathcal{X}|_{\max}}{\pi_{\min}}$. Thus, $\tilde{\mathcal{R}}_{M,\text{UCB}_4}(T) = O(\log^2 T)$.

(ii) If Δ_{\min} , $|\mathcal{X}|_{\max}$ and ρ_{\min} are unknown, choose a positive monotone sequences $\{\epsilon_t\}$ such that and $\{\kappa_t\}$ such that $\kappa_t \leq t$, $\epsilon_t \rightarrow 0$ and $\kappa_t \rightarrow \infty$ as $t \rightarrow \infty$. Then, $\tilde{\mathcal{R}}_{M,\text{UCB}_4}(T) = O(C(\epsilon_T)\kappa_T \log^2 T)$. We can choose $\{\epsilon_t\}$ and $\{\kappa_t\}$ as two arbitrarily slowly increasing sequences and thus the regret can be made arbitrarily close to $\log^2(T)$.

The proof follows by a combination of the proof of the theorems 3 and 4, and is omitted.

V. THE DECENTRALIZED MAB PROBLEM WITH I.I.D. REWARDS

We now consider the decentralized multi-armed bandit problem with i.i.d. rewards wherein multiple players play at the same time. Players have no information about means or distribution of rewards from various arms. There are no *dedicated control channels* for coordination or communication between the players. If two or more players pick the same arm, we assume that neither gets any reward. This is an online learning problem of distributed bipartite matching.

Distributed algorithms for bipartite matching algorithms are known [23], [24] which determine an ϵ -optimal matching with a ‘minimum’ amount of information exchange and computation. However, every run of this distributed bipartite matching algorithm incurs a cost due to computation, and communication necessary to exchange some information for decentralization. Let C be the cost per run, and $m(t)$ denote the number of times the distributed bipartite matching algorithm is run by time t . Then, under policy α the expected regret is

$$\mathcal{R}_\alpha(T) = T \sum_{i=1}^M \mu_{i,k_i^{**}} - \mathbb{E}_\alpha \left[\sum_{t=1}^T \sum_{i=1}^M X_{i,\alpha_i(t)}(t) \right] + C\mathbb{E}[m(T)]. \quad (27)$$

where k^{**} is the optimal matching as defined in equation (1) in section II-A.

Temporal Structure. We divide time into frames. Each frame is one of two kinds: a *decision frame*, and an *exploitation frame*. In the decision frame, the index is recomputed, and the distributed bipartite matching algorithm run again to determine the new matching. The length of

such a frame can be seen as cost of the algorithm. We further divide the decision frame into two phases, a *negotiation phase* and an *interrupt phase* (see Figure 1). The information exchange needed to compute an ϵ -optimal matching is done in the *negotiation phase*. In the *interrupt phase*, a player signals to other players if his allocation has changed. In the exploitation frame, the current matching is *exploited* without updating the indices. Later, we will allow the frame lengths to increase with time.

We now present the dUCB₄ algorithm, a decentralized version of UCB₄. For each player i and each arm j , we define a dUCB₄ index at the end of frame t as

$$g_{i,j}(t) := \bar{X}_{i,j}(t) + \sqrt{\frac{(M+2) \log n_i(t)}{n_{i,j}(t)}}, \quad (28)$$

where $n_i(t)$ is the number of successful plays (without collisions) of player i by frame t , $n_{i,j}(t)$ is the number of times player i picks arm j successfully by frame t . $\bar{X}_{i,j}(t)$ is the sample mean of rewards from arm j for player i from $n_{i,j}(t)$ samples. Let $g(t)$ denote the vector $(g_{i,j}(t), 1 \leq i \leq M, 1 \leq j \leq N)$. Note that g is computed only in the decision frames using the information available upto that time. Each player now uses the dUCB₄ algorithm. We will refer to an ϵ -optimal distributed bipartite matching algorithm as $\text{dBM}_\epsilon(g(t))$ that yields a solution $\mathbf{k}^*(t) := (k_1^*(t), \dots, k_M^*(t)) \in \mathcal{P}(N)$ such that $\sum_{i=1}^M g_{i,k_i^*(t)}(t) \geq \sum_{i=1}^M g_{i,k_i}(t) - \epsilon, \forall \mathbf{k} \in \mathcal{P}(N), \mathbf{k} \neq \mathbf{k}^*$. Let $\mathbf{k}^{**} \in \mathcal{P}(N)$ be such that $\mathbf{k}^{**} \in \arg \max_{\mathbf{k} \in \mathcal{P}(N)} \sum_{i=1}^M \mu_{i,k_i}$, i.e., an optimal bipartite matching with expected rewards from each matching. Denote $\mu^{**} := \sum_{i=1}^M \mu_{i,k_i^{**}}$, and define $\Delta_{\mathbf{k}} := \mu^{**} - \sum_{i=1}^M \mu_{i,k_i}, \mathbf{k} \in \mathcal{P}(N)$. Let $\Delta_{\min} = \min_{\mathbf{k} \in \mathcal{P}(N), \mathbf{k} \neq \mathbf{k}^{**}} \Delta_{\mathbf{k}}$ and $\Delta_{\max} = \max_{\mathbf{k} \in \mathcal{P}(N)} \Delta_{\mathbf{k}}$. We assume that $\Delta_{\min} > 0$.

In the dUCB₄ algorithm, at the end of every decision frame, the $\text{dBM}_\epsilon(g(t))$ will give a legitimate matching with no two players colliding on any arm. Thus, the regret accrues either if the matching $\mathbf{k}(t)$ is not the optimal matching \mathbf{k}^{**} , or if a decision frame is employed by the players to recompute the matching. Every time a frame is a decision frame, it adds a cost C to the regret. The cost C depends on two parameters: (a) the precision of the bipartite matching algorithm $\epsilon_1 > 0$, and (b) the precision of the index representation $\epsilon_2 > 0$. A bipartite matching algorithm has an ϵ_1 -precision if it gives an ϵ_1 -optimal matching. This would happen, for example, when

Algorithm 2 dUCB₄ for User i

```
1: Initialization: Play a set of matchings so that each player plays each arm at least once. Set counter  $\eta = 1$ .
2: while ( $t \leq T$ ) do
3:   if ( $\eta = 2^p$  for some  $p = 0, 1, 2, \dots$ ) then
4:     //Decision frame:
5:     Update  $g(t)$ ;
6:     Participate in the dBM $_{\epsilon}(g(t))$  algorithm to obtain a match  $k_i^*(t)$ ;
7:     if ( $k_i^*(t) \neq k_i^*(t-1)$ ) then
8:       Use interrupt phase to signal an INTERRUPT to all other players about changed allocation;
9:       Reset  $\eta = 1$ ;
10:    end if
11:    if (Received an INTERRUPT) then
12:      Reset  $\eta = 1$ ;
13:    end if
14:  else
15:    // Exploitation frame:
16:     $k_i^*(t) = k_i^*(t-1)$ ;
17:  end if
18:  Play arm  $k_i^*(t)$ ;
19:  Increment counter  $\eta = \eta + 1$ ,  $t = t + 1$ ;
20: end while
```

such an algorithm is run only for a finite number of rounds. The index has an ϵ_2 -precision if any two indices are not distinguishable if they are closer than ϵ_2 . This can happen for example when indices must be communicated to other players in a finite number of bits.

Thus, the cost C is a function of ϵ_1 and ϵ_2 , and can be denoted as $C(\epsilon_1, \epsilon_2)$, with $C(\epsilon_1, \epsilon_2) \rightarrow \infty$ as ϵ_1 or $\epsilon_2 \rightarrow 0$. Since, ϵ_1 and ϵ_2 are the parameters that are fixed *a priori*, we consider $\epsilon = \min(\epsilon_1, \epsilon_2)$ to specify both precisions. We denote the cost as $C(\epsilon)$.

We first show that if Δ_{\min} is known, we can choose an $\epsilon < \Delta_{\min}/(M+1)$, so that dUCB₄ algorithm will achieve order log-squared regret growth with T . If Δ_{\min} is not known, we can pick a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \rightarrow 0$, as $t \rightarrow \infty$. In a decentralized bipartite matching algorithm, the precision ϵ will depend on the amount of information exchanged in the decision frames. It, thus, is some monotonically decreasing function $\epsilon = f(L)$ of their length L such that $\epsilon \rightarrow 0$ as $L \rightarrow \infty$. Thus, we must pick a positive monotone sequence $\{L_t\}$ such that $L_t \rightarrow \infty$. Clearly, $C(f(L_t)) \rightarrow \infty$ as $t \rightarrow \infty$. This can happen arbitrarily slowly.

Theorem 6. (i) Let $\epsilon > 0$ be the precision of the bipartite matching algorithm and the precision of the index representation. If Δ_{\min} is known, choose $\epsilon > 0$ such that $\epsilon < \Delta_{\min}/(M+1)$. Let

L be the length of a frame. Then, the expected regret of the dUCB₄ algorithm is

$$\tilde{\mathcal{R}}_{\text{dUCB}_4}(T) \leq (L\Delta_{\max} + C(f(L))(1 + \log T)) \cdot \left(\frac{4M^3(M+2)N \log T}{(\Delta_{\min} - ((M+1)\epsilon)^2} + NM(2M+1) \right).$$

Thus, $\tilde{\mathcal{R}}_{\text{dUCB}_4}(T) = O(\log^2 T)$.

(ii) When Δ_{\min} is unknown, denote $\epsilon_{\min} = \Delta_{\min}/(2(M+1))$ and let $L_t \rightarrow \infty$ as $t \rightarrow \infty$. Then, there exists a $t_0 > 0$ such that for all $T > t_0$,

$$\begin{aligned} \tilde{\mathcal{R}}_{\text{dUCB}_4}(T) &\leq (L_{t_0}\Delta_{\max} + C(f(L_{t_0}))t_0 + (L_T\Delta_{\max} + C(f(L_T))(1 + \log T)) \cdot \\ &\quad \left(\frac{4M^3(M+2)N \log T}{(\Delta_{\min} - \epsilon_{\min})^2} + NM(2M+1) \right), \end{aligned}$$

where t_0 is the smallest t such that $f(L_{t_0}) < \epsilon_{\min}$. Thus by choosing an arbitrarily slowly increasing sequence $\{L_t\}$ we can make the regret arbitrarily close to $\log^2 T$.

Proof: (i) First, we obtain a bound for $L = 1$. Then, appeal to a result like Theorem 1 to obtain the result for general L . The implicit dependence between ϵ and L through the function $f(\cdot)$ does not affect this part of the analysis. Details are omitted due to space limitations.

We first upper bound the number of sub-optimal plays. We define $\tilde{n}_{i,j}(t)$, $1 \leq i \leq M, 1 \leq j \leq N$ as follows: Whenever the dBM _{ϵ} ($g(t)$) algorithm gives a non-optimal matching $\mathbf{k}(t)$, $\tilde{n}_{i,j}(t)$ is increased by one for some $(i, j) \in \arg \min_{1 \leq i \leq M, 1 \leq j \leq N} n_{i,j}(t)$. Let $\tilde{n}(T)$ denote the total number of suboptimal plays. Then, clearly, $\tilde{n}(T) = \sum_{i=1}^M \sum_{j=1}^N \tilde{n}_{i,j}(T)$. So, in order to get a bound on $\tilde{n}(T)$ we first get a bound on $\tilde{n}_{i,j}(T)$.

Let $\tilde{I}_{i,j}(t)$ be the indicator function which is equal to 1 if $\tilde{n}_{i,j}(t)$ is incremented by one, at time t . When $\tilde{I}_{i,j}(t) = 1$, there will be a corresponding matching $\mathbf{k}(t) \neq \mathbf{k}^{**}$ such that $k_i(t) = j$. In the following, we denote it as \mathbf{k} , omitting the time index. A non-optimal matching \mathbf{k} is selected if the event $\left\{ \sum_{i=1}^M g_{i,k_i^{**}}(m + 2^p - 1) \leq (M+1)\epsilon + \sum_{i=1}^M g_{i,k_i}(m + 2^p - 1) \right\}$ happens. If each index has an error of at most ϵ , the sum of M terms may introduce an error of at most $M\epsilon$. In addition, the distributed bipartite matching algorithm dBM _{ϵ} itself yields only an ϵ -optimal matching. This accounts for the term $(M+1)\epsilon$ above. Since the initial steps are similar to that

in Theorem 2, we skip those steps. Thus, similar to the equation (9), we get $\tilde{n}_{i,j}(T) \leq$

$$\begin{aligned}
l &+ \sum_{m=1}^T \sum_{p=0}^{\infty} 2^p I \left\{ \sum_{i=1}^M g_{i,k_i^{**}}(m+2^p-1) \leq (M+1)\epsilon + \sum_{i=1}^M g_{i,k_i}(m+2^p-1), \tilde{n}_{i,j}(m-1) \geq l \right\} \\
&\leq l + \sum_{m=1}^T \sum_{p=0}^{\infty} 2^p I \left\{ \sum_{i=1}^M \left(\bar{X}_{i,k_i^{**}}(m+2^p-1) + c_{m+2^p-1,n_{i,k_i^{**}}(m+2^p-1)} \right) \right. \\
&\quad \left. \leq (M+1)\epsilon + \sum_{i=1}^M \bar{X}_{i,k_i}(m+2^p-1) + c_{m+2^p-1,n_{i,k_i}(m+2^p-1)}, \tilde{n}_{i,j}(m-1) \geq l \right\} \\
&\leq l + \sum_{m=1}^T \sum_{p=0}^{\infty} 2^p I \left\{ \min_{1 \leq s_{1,k_1^{**}}, \dots, s_{M,k_M^{**}} < m+2^p} \sum_{i=1}^M \left(\bar{X}_{i,k_i^{**}}(m+2^p-1) + c_{m+2^p-1,s_{i,k_i^{**}}} \right) \right. \\
&\quad \left. \leq (M+1)\epsilon + \max_{l \leq s'_{1,k_1}, \dots, s'_{M,k_M} < m+2^p} \sum_{i=1}^M \left(\bar{X}_{i,k_i}(m+2^p-1) + c_{m+2^p-1,s'_{i,k_i}} \right) \right\} \\
&\leq l + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_{1,k_1^{**}}=1}^{m+2^p} \dots \sum_{s_{M,k_M^{**}}=1}^{m+2^p} \sum_{s'_{1,k_1}=1}^{m+2^p} \dots \sum_{s'_{M,k_M}=1}^{m+2^p} I \left\{ \sum_{i=1}^M \left(\bar{X}_{i,k_i^{**}}(m+2^p) + c_{m+2^p,s_{i,k_i^{**}}} \right) \right. \\
&\quad \left. \leq (M+1)\epsilon + \sum_{i=1}^M \left(\bar{X}_{i,k_i}(m+2^p) + c_{m+2^p,s'_{i,k_i}} \right) \right\}. \tag{29}
\end{aligned}$$

Now, it is easy to observe that the event

$$\left\{ \sum_{i=1}^M \left(\bar{X}_{i,k_i^{**}}(m+2^p) + c_{m+2^p,s_{i,k_i^{**}}} \right) \leq (M+1)\epsilon + \sum_{i=1}^M \left(\bar{X}_{i,k_i}(m+2^p) + c_{m+2^p,s'_{i,k_i}} \right) \right\}$$

implies at least one of the following events:

$$\begin{aligned}
A_i &:= \left\{ \bar{X}_{i,k_i^{**}}(m+2^p) \leq \mu_{i,k_i^{**}} - c_{m+2^p,s_{i,k_i^{**}}} \right\}, \\
B_i &:= \left\{ \bar{X}_{i,k_i}(m+2^p) \geq \mu_{i,k_i} + c_{m+2^p,s'_{i,k_i}} \right\}, 1 \leq i \leq M, \\
C &:= \left\{ \sum_{i=1}^M \mu_{i,k_i^{**}} < (M+1)\epsilon + \sum_{i=1}^M \mu_{i,k_i} + 2 \sum_{i=1}^M c_{m+2^p,s'_{i,k_i}} \right\} \\
D &:= \left\{ (M+1)\epsilon > \sum_{i=1}^M \mu_{i,k_i^{**}} - \sum_{i=1}^M \mu_{i,k_i} \right\}. \tag{30}
\end{aligned}$$

Using the Chernoff-Hoeffding inequality, we get $\mathbb{P}(A_i) \leq (m+2^p)^{-2(M+2)}$, $\mathbb{P}(B_i) \leq (m+2^p)^{-2(M+2)}$, $1 \leq i \leq M$. For $l \geq \left\lceil \frac{4M^2(M+2)\log T}{(\Delta_{\min} - (M+1)\epsilon)^2} \right\rceil$, we get

$$\begin{aligned}
& \sum_{i=1}^M \mu_{i,k_i^{**}} - \sum_{i=1}^M \mu_{i,k_i} - (M+1)\epsilon - 2 \sum_{i=1}^M c_{m+2^p, s'_{i,k_i}} \\
& \geq \sum_{i=1}^M \mu_{i,k_i^{**}} - \sum_{i=1}^M \mu_{i,k_i} - (M+1)\epsilon - 2M \sqrt{\frac{(M+2) \log(m+2^p)}{l}} \\
& \geq \sum_{i=1}^M \mu_{i,k_i^{**}} - \sum_{i=1}^M \mu_{i,k_i} - (M+1)\epsilon - (\Delta_{min} - (M+1)\epsilon) \geq 0
\end{aligned} \tag{31}$$

The event D is false by assumption. So, we get, $\mathbb{E}[\tilde{n}_{i,j}(T)]$

$$\begin{aligned}
& \leq \left\lceil \frac{4M^2(M+2) \log T}{(\Delta_{min} - (M+1)\epsilon)^2} \right\rceil + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_{1,k_1^{**}}=1}^{m+2^p} \dots \sum_{s_{M,k_M^{**}}=1}^{m+2^p} \sum_{s'_{1,k_1}=1}^{m+2^p} \dots \sum_{s'_{M,k_M}=1}^{m+2^p} 2M(m+2^p)^{-2(M+2)} \\
& \leq \left\lceil \frac{4M^2(M+2) \log T}{(\Delta_{min} - (M+1)\epsilon)^2} \right\rceil + 2M \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m+2^p)^{-4} \\
& \leq \frac{4M^2(M+2) \log T}{(\Delta_{min} - (M+1)\epsilon)^2} + (2M+1).
\end{aligned} \tag{32}$$

Now, putting it all together, we get

$$\mathbb{E}[\tilde{n}(T)] = \sum_{i=1}^M \sum_{j=1}^N \mathbb{E}[\tilde{n}_{i,j}(T)] \leq \frac{4M^3(M+2)N \log T}{(\Delta_{min} - (M+1)\epsilon)^2} + (2M+1)MN.$$

Now, by the proof of Theorem 2 (c.f. equation(15), $\mathbb{E}[m(T)] \leq \mathbb{E}[\tilde{n}(T)](1 + \log T)$. We can now

bound the regret, $\tilde{\mathcal{R}}_{\text{dUCB}_4}(T) = \sum_{k \in \mathcal{P}(N), k \neq k^{**}} \Delta_k \sum_{i=1}^M \mathbb{E}[\tilde{n}_{i,k_i}(T)] + C\mathbb{E}[m(T)]$

$$\begin{aligned}
& \leq \Delta_{max} \sum_{k \in \mathcal{P}(N), k \neq k^{**}} \sum_{i=1}^M \mathbb{E}[\tilde{n}_{i,k_i}(T)] + C\mathbb{E}[m(T)] \\
& = \Delta_{max} \mathbb{E}[\tilde{n}(T)] + C\mathbb{E}[m(t)].
\end{aligned}$$

For a general L , by Theorem 1 we get

$$\tilde{\mathcal{R}}_{\text{dUCB}_4}(T) \leq L\Delta_{max} \mathbb{E}[\tilde{n}(T)] + C(f(L))\mathbb{E}[m(T)] \leq (L\Delta_{max} + C(f(L))(1 + \log T))\mathbb{E}[\tilde{n}(T)].$$

Now, using the bound (33), we get the desired upper bound on the expected regret.

(ii) Since $\epsilon_t = f(L_t)$ is a monotonically decreasing function of L_t such that $\epsilon_t \rightarrow 0$ as $L_t \rightarrow \infty$, there exists a t_0 such that for $t > t_0$, $\epsilon_t < \epsilon_{min}$. We may get a linear regret upto time t_0 but after that by the analysis of Theorem 2, regret grows only sub-linearly. Since $C(\cdot)$ is monotonically

increasing, $C(f(L_T)) \geq C(f(L_t)), \forall t \leq T$, we get the desired result. The last part is illustrative and can be trivially established using the obtained bound on the regret in (ii). ■

Remarks. 1. We note that in the initial steps, our proof followed the proof of the main result in [12].

2. The UCB₂ algorithm described in [4] performs computations only at exponentially spaced time epochs. So, it is natural to imagine that a decentralized algorithm based on it could be developed, and get a better regret bound. Unfortunately, the single player UCB₂ algorithm has an obvious weakness: regret is linear in the number of arms. Thus, the decentralized/combinatorial extension of UCB₂ would yield regret growing exponentially in the number of players and arms. We use a similar index but a different scheme, allowing us to achieve *poly-log* regret growth and a linear memory requirement for each player.

VI. THE DECENTRALIZED MAB PROBLEM WITH MARKOVIAN REWARDS

Now, we consider the decentralized MAB problem with M players and N arms where the rewards obtained each time when an arm is pulled are not i.i.d. but come from a Markov chain. The reward that player i gets from arm j (when there is no collision) X_{ij} , is modelled as an irreducible, aperiodic, reversible Markov chain on a finite state space $\mathcal{X}^{i,j}$ and represented by a transition probability matrix $P^{i,j} := (p_{x,x'}^{i,j} : x, x' \in \mathcal{X}^{i,j})$. Assume that $\mathcal{X}^{i,j} \in (0, 1]$. Let $X_{i,j}(1), X_{i,j}(2), \dots$ denote the successive rewards from arm j for player i . All arms are mutually independent for all players. Let $\pi^{i,j} := (\pi_x^{i,j}, x \in \mathcal{X}^{i,j})$ be the stationary distribution of the Markov chain $P^{i,j}$. The mean reward from arm j for player i is defined as $\mu_{i,j} := \sum_{x \in \mathcal{X}^{i,j}} x \pi_x^{i,j}$. Note that the Markov chain represented by $P^{i,j}$ makes a state transition only when player i plays arm j . Otherwise, it remains *rested*. As described in the previous section, $n_i(t)$ is the number of successful plays (without collisions) of player i by frame t , $n_{i,j}(t)$ is the number of times player i picks arm j successfully by frame t and $\bar{X}_{i,j}(t)$ is the sample mean of rewards from arm j for player i from $n_{i,j}(t)$ samples. Denote $\Delta_{\min} := \min_{\mathbf{k} \in \mathcal{P}(N), \mathbf{k} \neq \mathbf{k}^{**}} \Delta_{\mathbf{k}}$ and $\Delta_{\max} := \max_{\mathbf{k} \in \mathcal{P}(N)} \Delta_{\mathbf{k}}$. Denote $\pi_{\min} := \min_{1 \leq i \leq M, 1 \leq j \leq N, x \in \mathcal{X}^{i,j}} \pi_x^{i,j}$, $x_{\max} := \max_{1 \leq i \leq M, 1 \leq j \leq N, x \in \mathcal{X}^{i,j}} x$ and $x_{\min} := \min_{1 \leq i \leq M, 1 \leq j \leq N, x \in \mathcal{X}^{i,j}} x$. Let $\hat{\pi}_x^{i,j} := \max\{\pi_x^{i,j}, 1 - \pi_x^{i,j}\}$ and $\hat{\pi}_{\max} := \max_{1 \leq i \leq M, 1 \leq j \leq N, x \in \mathcal{X}^{i,j}} \hat{\pi}_x^{i,j}$. Let $|\mathcal{X}^{i,j}|$ denote the cardinality of the state space $\mathcal{X}^{i,j}$, $|\mathcal{X}|_{\max} := \max_{1 \leq i \leq M, 1 \leq j \leq N} |\mathcal{X}^{i,j}|$. Let

$\rho^{i,j}$ be the eigenvalue gap, $1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the matrix P^{i,j^2} .

Denote $\rho_{max} := \max_{1 \leq i \leq M, 1 \leq j \leq N} \rho^{i,j}$ and $\rho_{min} := \min_{1 \leq i \leq M, 1 \leq j \leq N} \rho^{i,j}$.

The total reward obtained by time T is $S_T = \sum_{j=1}^N \sum_{i=1}^M \sum_{s=1}^{n_{i,j}(T)} X_{i,j}(s)$ and the regret is

$$\tilde{\mathcal{R}}_{M,\alpha}(T) := T \sum_{i=1}^M \mu_{i,k_i^{**}} - \mathbb{E}_\alpha \left[\sum_{j=1}^N \sum_{i=1}^M \sum_{s=1}^{n_{i,j}(T)} X_{i,j}(s) \right] + C\mathbb{E}[m(T)]. \quad (33)$$

Define the index

$$g_{i,j}(t) := \bar{X}_{i,j}(t) + \sqrt{\frac{\kappa \log n_i(t)}{n_{i,j}(t)}} \quad (34)$$

where κ be any constant such that $\kappa > (112 + 56M)|\mathcal{X}|_{max}^2/\rho_{min}$.

We need the following lemma to prove the regret bound.

Lemma 2. *If the reward of each player-arm pair (i, j) is given by a Markov chain, satisfying the properties of Lemma 4, then under any policy α*

$$\tilde{\mathcal{R}}_{M,\alpha}(T) \leq \sum_{k \in \mathcal{P}(N), k \neq k^{**}} \Delta_k \mathbb{E}[n^k(T)] + C\mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P} \quad (35)$$

where $n^k(T)$ is the number of times that the matching k occurred by the time T and $\tilde{K}_{\mathcal{X},P}$ is defined as $\tilde{K}_{\mathcal{X},P} = \sum_{j=1}^N \sum_{i=1}^M \sum_{x \in \mathcal{X}^{i,j}} x/\pi_{min}^j$

Proof: Let $(X_{i,j}(1), X_{i,j}(2), \dots)$ denote the successive rewards for player i from arm j . Let $\mathcal{F}_t^{i,j}$ denote the σ -algebra generated by $(X_{i,j}(1), \dots, X_{i,j}(t))$, $\mathcal{F}^{i,j} = \vee_{t \geq 1} \mathcal{F}_t^{i,j}$ and $\mathcal{G}^{i,j} = \vee_{(k,l) \neq (i,j)} \mathcal{F}^{k,l}$. Since arms are independent, $\mathcal{G}^{i,j}$ is independent of $\mathcal{F}^{i,j}$. Clearly, $n_{i,j}(T)$ is a stopping time with respect to $\mathcal{F}^{i,j} \vee \mathcal{G}_T^{i,j}$. The total reward is

$$S_T = \sum_{j=1}^N \sum_{i=1}^M \sum_{t=1}^{n_{i,j}(T)} X_{i,j}(t) = \sum_{j=1}^N \sum_{i=1}^M \sum_{x \in \mathcal{X}^{i,j}} x N(x, n_{i,j}(T))$$

where $N(x, n_{i,j}(T)) := \sum_{t=1}^{n_{i,j}(T)} I\{X_{i,j}(t) = x\}$. Taking expectations and using the Lemma 4,

$$\left| \mathbb{E}[S_T] - \sum_{j=1}^N \sum_{i=1}^M \sum_{x \in \mathcal{X}^{i,j}} x \pi_x^{i,j} \mathbb{E}[n_{i,j}(T)] \right| \leq \sum_{j=1}^N \sum_{i=1}^M \sum_{x \in \mathcal{X}^{i,j}} x/\pi_{min}^{i,j}$$

which implies,

$$\left| \mathbb{E}[S_T] - \sum_{j=1}^N \sum_{i=1}^M \mu_{i,j} \mathbb{E}[n_{i,j}(T)] \right| \leq \tilde{K}_{\mathcal{X},P}$$

where $\tilde{K}_{\mathcal{X},P} = \sum_{j=1}^N \sum_{i=1}^M \sum_{x \in \mathcal{X}^{i,j}} x / \pi_{min}^{i,j}$. Now,

$$\begin{aligned} \sum_{j=1}^N \sum_{i=1}^M \mu_{i,j} \mathbb{E}[n_{i,j}(T)] &= \sum_{j=1}^N \sum_{i=1}^M \sum_{k \in \mathcal{P}(N), (i,j) \in k} \mu_{i,k_i} \mathbb{E}[n_{i,k_i}(T)] = \sum_{k \in \mathcal{P}(N)} \sum_{i=1}^M \mu_{i,k_i} \mathbb{E}[n_{i,k_i}(T)] \\ &= \sum_{k \in \mathcal{P}(N)} \mu^k \mathbb{E}[n^k(T)] \end{aligned}$$

where $\mu^k = \sum_{i=1}^M \mu_{i,k_i}$. Since regret is defined as in the equation (33),

$$\left| \tilde{\mathcal{R}}_{M,\alpha}(T) - \left(T\mu^{**} - \sum_{k \in \mathcal{P}(N), (i,j) \in k} \mu_{i,k_i} \mathbb{E}[n_{i,k_i}(T)] + C\mathbb{E}_\alpha[m(T)] \right) \right| \leq \tilde{K}_{\mathcal{X},P}. \quad (36)$$

■

The main result of this section is the following.

Theorem 7. (i) Let $\epsilon > 0$ be the precision of the bipartite matching algorithm and the precision of the index representation. If Δ_{min} , $|\mathcal{X}|_{max}$ and ρ_{min} are known, choose $\epsilon > 0$ such that $\epsilon < \Delta_{min}/(M+1)$ and $\kappa > (112 + 56M)|\mathcal{X}|_{max}^2/\rho_{min}$. Let L be the length of a frame. Then, the expected regret of the dUCB₄ algorithm with index (34) for the decentralized MAB problem with Markovian rewards and per computation cost C is given by

$$\begin{aligned} &\tilde{\mathcal{R}}_{M,\text{dUCB}_4}(T) \\ &\leq (L\Delta_{max} + C(f(L))(1 + \log T)) \cdot \left(\frac{4M^3\kappa N \log T}{(\Delta_{min} - (M+1)\epsilon)^2} + (2MD + 1)MN \right) + \tilde{K}_{\mathcal{X},P}. \end{aligned}$$

Thus, $\tilde{\mathcal{R}}_{M,\text{dUCB}_4}(T) = O(\log^2 T)$.

(ii) If Δ_{min} , $|\mathcal{X}|_{max}$ and ρ_{min} are unknown, denote $\epsilon_{min} = \Delta_{min}/(2(M+1))$ and let $L_t \rightarrow \infty$ as $t \rightarrow \infty$. Also, choose a positive monotone sequence $\{\kappa_t\}$ such that $\kappa_t \rightarrow \infty$ as $t \rightarrow \infty$ and $\kappa_t \leq t$. Then, $\tilde{\mathcal{R}}_{M,\text{dUCB}_4}(T) = O(C(f(L_T))\kappa_T \log^2 T)$. Thus by choosing an arbitrarily-slowly increasing sequences, we can make the regret arbitrarily close to $\log^2 T$.

Proof: (i) We skip the initial steps as they are same as in the proof of Theorem 6. We start

by bounding $\tilde{n}_{i,j}(T)$ as defined in the proof of Theorem 6. Then, from equation (29), we get

$$\begin{aligned} &\leq l + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_{1,k_1}^{**}=1}^{m+2^p} \dots \sum_{s_{M,k_M}^{**}=1}^{m+2^p} \sum_{s'_{1,k_1}=1}^{m+2^p} \dots \sum_{s'_{M,k_M}=1}^{m+2^p} I\left\{\sum_{i=1}^M \left(\bar{X}_{i,k_i}^{**}(m+2^p) + c_{m+2^p,s_{i,k_i}^{**}}\right)\right. \\ &\quad \left.\leq (M+1)\epsilon + \sum_{i=1}^M \left(\bar{X}_{i,k_i}(m+2^p) + c_{m+2^p,s'_{i,k_i}}\right)\right\} \end{aligned} \quad (37)$$

Now, the event in the parenthesis $\{\cdot\}$ above implies at least one of the events (A_i, B_i, C, D) given in the display (30). From the proof of Theorem 4 (equations (22, 23), $\mathbb{P}(A_i) \leq D(m+2^p)^{-\frac{\kappa\rho_{\min}}{28|\mathcal{X}|_{\max}^2}}$, $\mathbb{P}(B_i) \leq D(m+2^p)^{-\frac{\kappa\rho_{\min}}{28|\mathcal{X}|_{\max}^2}}$, $1 \leq i \leq M$. Similar to the steps in display (31), we can show that the event C is false. Also, the event D is false by assumption. So, similar to the proof of the Theorem 6 (c.f. display (32) we get,

$$\begin{aligned} \mathbb{E}[\tilde{n}_{i,j}(T)] &\leq \left\lceil \frac{4M^2\kappa \log T}{(\Delta_{\min} - (M+1)\epsilon)^2} \right\rceil + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_{1,k_1}^{**}=1}^{m+2^p} \dots \sum_{s_{M,k_M}^{**}=1}^{m+2^p} \\ &\quad \sum_{s'_{1,k_1}=1}^{m+2^p} \dots \sum_{s'_{M,k_M}=1}^{m+2^p} 2MD(m+2^p)^{-\frac{\kappa\rho_{\min}}{28|\mathcal{X}|_{\max}^2}} \\ &\leq \left\lceil \frac{4M^2\kappa \log T}{(\Delta_{\min} - (M+1)\epsilon)^2} \right\rceil + 2MD \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m+2^p)^{-\frac{\kappa\rho_{\min} - 56M|\mathcal{X}|_{\max}^2}{28|\mathcal{X}|_{\max}^2}} \\ &\leq \frac{4M^2\kappa \log T}{(\Delta_{\min} - (M+1)\epsilon)^2} + (2MD + 1). \end{aligned}$$

when $\kappa > (112 + 56M)|\mathcal{X}|_{\max}^2/\rho_{\min}$. Now, putting it all together, we get

$$\mathbb{E}[\tilde{n}(T)] = \sum_{i=1}^M \sum_{j=1}^N \mathbb{E}[\tilde{n}_{i,j}(T)] \leq \frac{4M^3\kappa N \log T}{(\Delta_{\min} - (M+1)\epsilon)^2} + (2MD + 1)MN.$$

Now, by proof of the Theorem 2 (equation (15)), $\mathbb{E}[m(T)] \leq \mathbb{E}[\tilde{n}(T)](1 + \log T)$. We can now bound the regret,

$$\begin{aligned} \tilde{\mathcal{R}}_{M,\text{dUCB}_4}(T) &= \sum_{k \in \mathcal{P}(N), k \neq k^{**}} \Delta_k \sum_{i=1}^M \mathbb{E}[\tilde{n}_{i,k_i}(T)] + C\mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P} \\ &\leq \Delta_{\max} \sum_{k \in \mathcal{P}(N), k \neq k^{**}} \sum_{i=1}^M \mathbb{E}[\tilde{n}_{i,k_i}(T)] + C\mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P} \end{aligned}$$

$$= \Delta_{max} \mathbb{E}[\tilde{n}(T)] + C \mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P}.$$

For a general L , by Theorem 1

$$\begin{aligned} \tilde{\mathcal{R}}_{M, \text{dUCB}_4}(T) &\leq L \Delta_{max} \mathbb{E}[\tilde{n}(T)] + C(f(L)) \mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P}. \\ &\leq (L \Delta_{max} + C(f(L))(1 + \log T)) \mathbb{E}[\tilde{n}(T)] + \tilde{K}_{\mathcal{X},P}. \end{aligned}$$

Now, using the bound (38), we get the desired upper bound on the expected regret.

(ii) This can now easily be obtained using the above and following Theorem 6. ■

VII. DISTRIBUTED BIPARTITE MATCHING: ALGORITHM AND IMPLEMENTATION

In the previous section, we referred to an unspecified distributed algorithm for bipartite matching dBM, that is used by the dUCB₄ algorithm. We now present one such algorithm, namely, Bertsekas' auction algorithm [17], and its distributed implementation. We note that the presented algorithm is not the only one that can be used. The dUCB₄ algorithm will work with a distributed implementation of any bipartite matching algorithm, e.g. algorithms given in [24].

Consider a bipartite graph with M players on one side, and N arms on the other, and $M \leq N$. Each player i has a value $\mu_{i,j}$ for each arm j . Each player knows only his own values. Let us denote by k^{**} , a matching that maximizes the matching surplus $\sum_{i,j} \mu_{i,j} x_{i,j}$, where the variable $x_{i,j}$ is 1 if i is matched with j , and 0 otherwise. Note that $\sum_i x_{i,j} \leq 1, \forall j$, and $\sum_j x_{i,j} \leq 1, \forall i$. Our goal is to find an ϵ -optimal matching. We call any matching k^* to be ϵ -optimal if $\sum_i \mu_{i,k^{**}(i)} - \sum_i \mu_{i,k^*(i)} \leq \epsilon$.

Algorithm 3 : dBM _{ϵ} (Bertsekas Auction Algorithm)

- 1: All players i initialize prices $p_j = 0, \forall$ channels j ;
 - 2: **while** (prices change) **do**
 - 3: Player i communicates his preferred arm j_i^* and bid $b_i = \max_j (\mu_{ij} - p_j) - 2\max_j (\mu_{ij} - p_j) + \frac{\epsilon}{M}$ to all other players.
 - 4: Each player determines on his own if he is the *winner* i_j^* on arm j ;
 - 5: All players set prices $p_j = \mu_{i_j^*,j}$;
 - 6: **end while**
-

Here, $2\max_j$ is the second highest maximum over all j . The *best* arm for a player i is arm $j_i^* = \arg \max_j (\mu_{i,j} - p_j)$. The *winner* i_j^* on an arm j is the one with the highest bid.

The following lemma in [17] establishes that Bertsekas' auction algorithm will find the ϵ -optimal matching in a finite number of steps.

Lemma 3. [17] *Given $\epsilon > 0$, Algorithm 3 with rewards $\mu_{i,j}$, for player i playing the j th arm, converges to a matching k^* such that $\sum_i \mu_{i,k^{**}(i)} - \sum_i \mu_{i,k^*(i)} \leq \epsilon$ where k^{**} is an optimal matching. Furthermore, this convergence occurs in less than $(M^2 \max_{i,j} \{\mu_{i,j}\})/\epsilon$ iterations.*

The temporal structure of the dUCB₄ algorithm is such that time is divided into frames of length L . Each frame is either a *decision* frame, or an *exploitation* frame. In the exploitation frame, each player plays the arm it was allocated in the last decision frame. The distributed bipartite matching algorithm (e.g. based on Algorithm 3), is run in the decision frame. The decision frame has an interrupt phase of length M and negotiation phase of length $L - M$. We now describe an implementation structure for these phases in the decision frame.

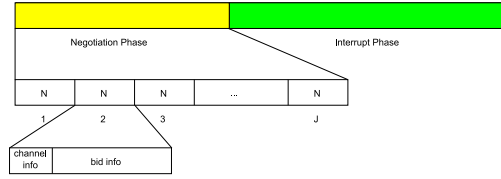


Fig. 1. Structure of the decision frame

Interrupt Phase: The interrupt phase can be implemented very easily. It has length M time slots. On a pre-determined channel, each player by turn transmits a '1' if the arm with which it is now matched has changed, '0' otherwise. If any user transmits a '1', everyone knows that the matching has changed, and they reset their counter $\eta = 1$.

Negotiation Phase: The information needed to be exchanged to compute an ϵ -optimal matching is done in the negotiation phase. We first provide a *packetized implementation* of the negotiation phase. The negotiation phase consists of J subframes of length M each (See figure 1). In each subframe, the users transmit a packet by turn. The packet contains bid information: (channel number, bid value). Since all users transmit by turn, all the users know the bid values by the end of the subframe, and can compute the new allocation, and the prices independently. The length of the subframe J determines the precision ϵ of the distributed bipartite matching algorithm. Note that in the packetized implementation, $\epsilon_1 = 0$, i.e., bid values can be computed exactly,

and for a given ϵ_2 , we can determine J , the number of rounds the dBM algorithm 3 runs for, and returns an ϵ_2 -optimal matching.

If a packetized implementation is not possible, we can give a *physical implementation*. Our only assumption here is going to be that each user can observe a channel, and determine if there was a successful transmission on it, a collision, or no transmission, in a given time slot. The whole negotiation phase is again divided into J sub-frames. In each sub-frame, each user transmits by turn. It simply transmits $\lceil \log M \rceil$ bits to indicate a channel number, and then $\lceil \log 1/\epsilon_1 \rceil$ bits to indicate its bid value to precision ϵ_1 . The number of such sub-frames J is again chosen so that the dBM algorithm (based on Algorithm 3) returns an ϵ_2 -optimal matching.

VIII. SIMULATIONS

We illustrate the empirical performance of the dUCB₄ algorithm when the successive rewards from a channel are i.i.d. and when they are Markovian. Consider two users and two channels. In the i.i.d. case, each channel has rewards that are generated by a Bernoulli distribution taking values 0 and 1. The first user has mean rewards of 0.8 and 0.6 for channels 1 and 2 respectively. The second user has mean rewards of 0.6 and 0.35. The algorithm's performance, averaged over 50 runs, is shown in Figure 2 (i). It shows cumulative regret with time. The red bold curve is the theoretical upper bound we derived, while the blue curve is the observed regret. The algorithm seems to perform much better than even the poly-log regret upper bound we derived.

In the Markovian case, rewards are generated by a Markov chain having states 0 and 1. The mean reward on a channel is given by its stationary distribution, i.e., the probability the Markov chain is in state 1, π_1 . The properties of the Markov chains are given in Table I. The performance of the dUCB₄ algorithm on this model, averaged over 50 runs, is shown in Figure 2 (ii). Once again, the algorithm seems to perform much better than even the poly-log regret upper bound we derived.

IX. CONCLUSIONS

We have proposed a dUCB₄ algorithm for decentralized learning in multi-armed bandit problems that achieves a regret of near- $O(\log^2(T))$. Finding a lower bound is usually quite difficult, and

TABLE I
MARKOV CHAIN PARAMETERS : TRANSITION PROBABILITY AND STATIONARY DISTRIBUTION

User	Channel	p_{01}, p_{10}	π
1	1	0.3, 0.5	0.3/0.8
1	2	0.2, 0.6	0.2/0.8
2	1	0.6, 0.3	0.6/0.9
2	2	0.7, 0.2	0.7/0.9

currently a work in progress.

REFERENCES

- [1] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4-22, 1985.
- [2] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multi-armed bandit problem with multiple plays - part i: i.i.d. rewards," *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 968-975, November, 1987.
- [3] R. Agrawal, "Sample mean based index policies with $(O(\log n))$ regret for the multi-armed bandit problem," *Advances in Applied Probability*, Vol. 27, No. 4, pp. 1054-1078, 1995.
- [4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235-256, 2002.
- [5] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multi-armed bandit problem with multiple plays - part ii: Markovian rewards," *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 977-982, November 1987.
- [6] C. Tekin and M. Liu, "Online algorithms for the multi-armed bandit problem with markovian rewards," *Allerton Conference on Communication, Control, and Computing*, October, 2010.
- [7] C. Papadimitriou and J. Tsitsiklis, "The complexity of optimal queuing network control," *Mathematics of Operations Research*, vol. 24, no. 2, pp. 293-305, May, 1999.
- [8] C. Tekin and M. Liu, "Online learning in opportunistic spectrum access: A restless bandit approach," *International Conference on Computer Communications (INFOCOM), Shanghai, China.*, April 2011.
- [9] W. Dai, Y. Gai, and B. Krishnamachari, "Efficient online learning for opportunistic spectrum access," *International Conference on Computer Communications (INFOCOM), Mini Conference, Orlando, USA*, March, 2012.
- [10] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Restless multi-armed bandit with unknown dynamics," *IEEE Transactions on Information Theory*, Submitted, November, 2011.
- [11] W. Dai, Y. Gai, B. Krishnamachari, and Q. Zhao, "The non-bayesian restless multi-armed bandit: A case of near-logarithmic regret," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May, 2011.
- [12] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Trans. on Networking*, to appear, 2012.

- [13] Y. Gai, B. Krishnamachari, and M. Liu, "On the combinatorial multi-armed bandit problem with markovian rewards," *IEEE Global Communications Conference (GLOBECOM)*, December, 2011.
- [14] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5667-5681, November, 2010.
- [15] A. Anandkumar, N. Michael, A. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE JSAC on Advances in Cognitive Radio Networking and Communications*, April, 2011.
- [16] Y. Gai and B. Krishnamachari, "Decentralized online learning algorithms for opportunistic spectrum access," *IEEE Global Communications Conference (GLOBECOM 2011)*, December, 2011.
- [17] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Computational Optimization and Applications*, vol. 1, pp. 7-66, 1992.
- [18] E. Hossain and V. K. Bhargava, "Cognitive wireless communication networks," *Springer*, 2007.
- [19] D. Pollard, "Convergence of stochastic processes," *Springer*, 1984.
- [20] K. Liu and Q. Zhao, "Multi-armed bandit problems with heavy tail reward distributions," *Allerton Conference on Communication, Control, and Computing*, September, 2011.
- [21] P. Lezaud, "Chernoff-type bound for finite markov chains," *Ann. Appl. Prob.*, vol. 8, pp. 849-867, 1998.
- [22] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. on Information Theory*, Submitted, 2012.
- [23] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Annals of Operations Research*, vol. 14, 1988.
- [24] M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," *Proceedings of the IEEE Conference on Decision and Control*, December, 2008.

APPENDIX

Let $(X_t, t = 1, 2, \dots)$ be an irreducible, aperiodic and reversible Markov chain on a finite state space \mathcal{X} with transition probability matrix P , a stationary distribution π and an initial distribution λ . Let \mathcal{F}_t be the σ -algebra generated by (X_1, X_2, \dots, X_t) . Denote $N_\lambda = \left\| \left(\frac{\lambda_x}{\pi_x}, x \in \mathcal{X} \right) \right\|_2$.

Lemma 4. [5] *Let \mathcal{G} be a σ -algebra independent of $\mathcal{F} = \vee_{t \geq 1} \mathcal{F}_t$. Let τ be a stopping time of $\mathcal{F}_t \vee \mathcal{G}$. Let $N(x, \tau) := \sum_{t=1}^{\tau} I\{X_t = x\}$. Then, $|\mathbb{E}[N(x, \tau)] - \pi_x \mathbb{E}[\tau]| \leq K$, where $K \leq 1/\pi_{\min}$ and $\pi_{\min} = \min_{x \in \mathcal{X}} \pi_x$. K depends on P .*

Lemma 5. [21] *Denote $N_\lambda = \left\| \left(\frac{\lambda_x}{\pi_x}, x \in \mathcal{X} \right) \right\|_2$. Let ρ be the eigenvalue gap, $1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the matrix P^2 . Let $f : \mathcal{X} \rightarrow \mathbb{R}$ be such that $\sum_{x \in \mathcal{X}} \pi_x f(x) = 0$, $\|f\|_\infty \leq 1$, $\|f\|_2^2 \leq 1$. Then, for any $\gamma > 0$, $\mathbb{P} \left(\sum_{a=1}^t f(X_a)/t \geq \gamma \right) \leq N_\lambda e^{-t\rho\gamma^2/28}$.*

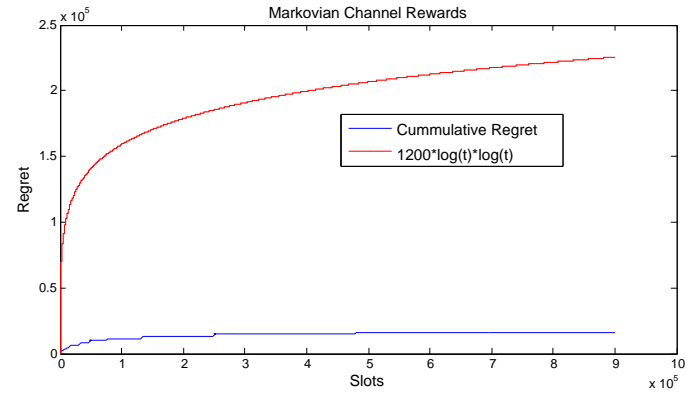
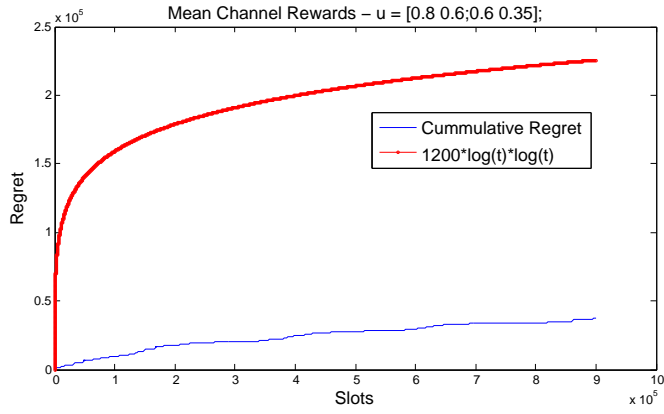


Fig. 2. (i) Cumulative regret : 2 users, 2 channels; i.i.d. channels; Mean reward matrix = $[0.8, 0.6; 0.6, 0.35]$. (ii) Cumulative regret : 2 users, 2 channels; Markovian channels.